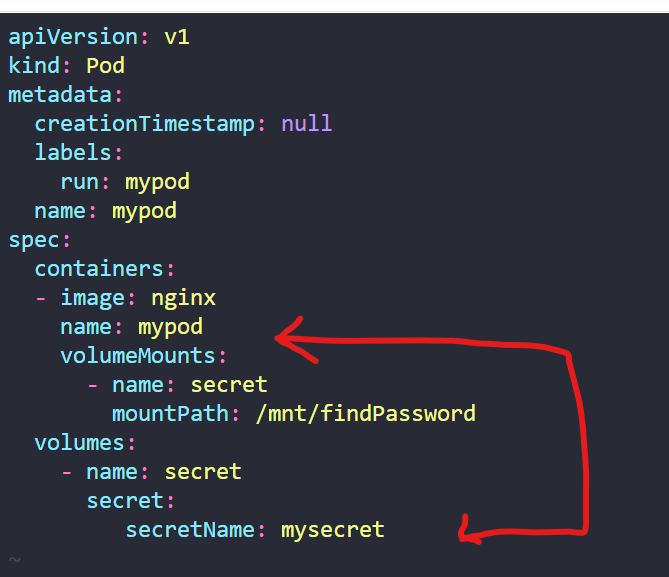
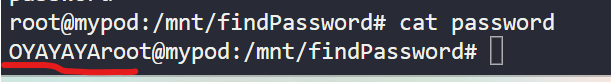
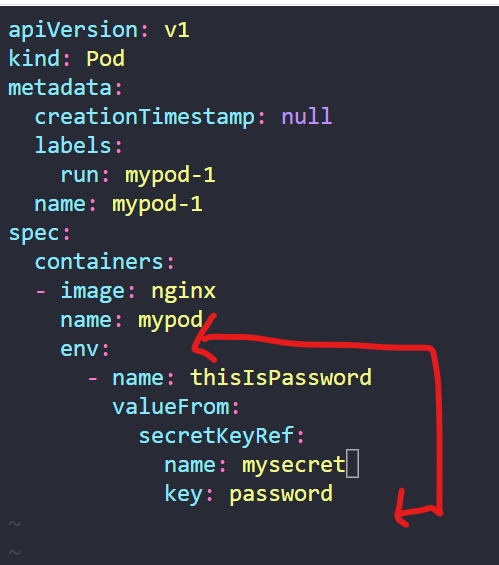
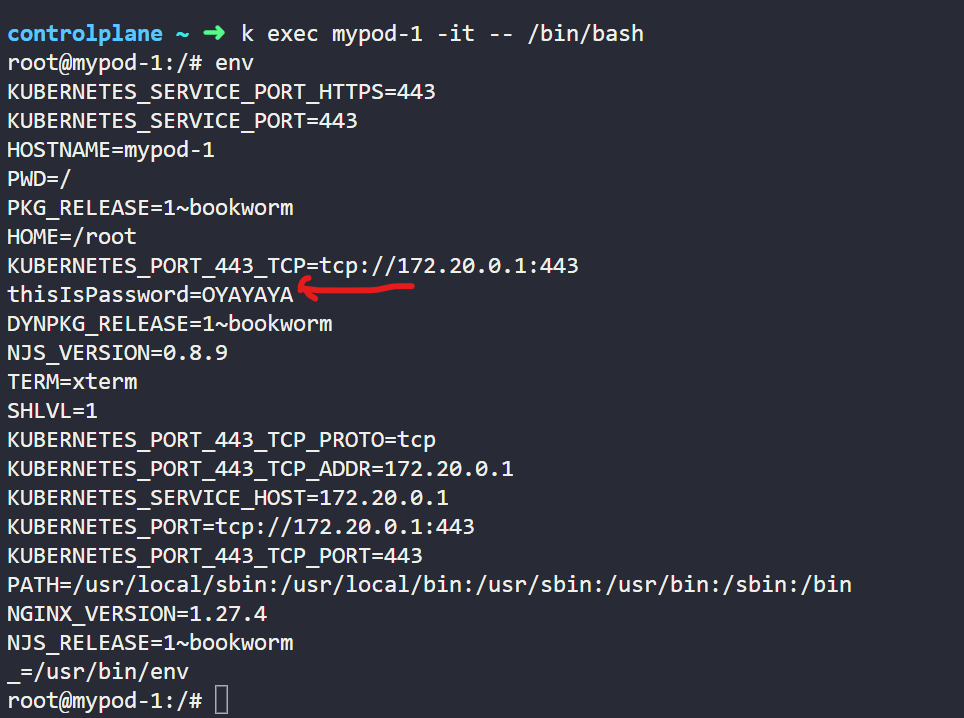
**Secrets from Kubernets**

1. Simple (opaque) secrets can be used two ways in a pod.
   1. As a volume Mount, where Kubernetes will store the data inside the container at the given path as a file. [Note each secret will be generated as a separate file]
   2. As environment variable, where Kubernetes will simply make the secrets as environment variables
2. So first create a secret using below command:
   1. k create secret generic mysecret --from-literal=password=OYAYAYA
3. After that, create a pod using below commands, in first time we will use secret as a volume Mount:
   1. k run mypod --image=nginx --dry-run=client -o yaml > pod.yaml
   2. add below volume section in mypod 
   3. k apply -f pod.yaml
4. Now go into the pod and check if the password has been created:
   1. K exec mypod -it -- /bin/bash
   2. Cat /mnt/findPassword/password 
5. Now lets use the same secret as environment variable
6. Now make changes in the same pod.yaml 
7. Now check the environment variables from inside the pod: 
8. Other types of secrets:

Here's a concise summary on types of secrets:

* **Opaque:**  
  Default secret type for arbitrary key/value pairs (base64-encoded). Use when no specific format is needed.
* **kubernetes.io/tls:**  
  Stores TLS certificate (tls.crt) and key (tls.key). Use for securing communications.
* **Docker Credentials (kubernetes.io/dockerconfigjson / dockercfg):**  
  Holds container registry credentials. Use for authenticating to Docker registries.
* **kubernetes.io/basic-auth:**  
  Stores username and password for basic HTTP authentication.
* **kubernetes.io/ssh-auth:**  
  Contains SSH keys for secure shell operations.
* **kubernetes.io/service-account-token:**  
  Auto-generated for service accounts to access the Kubernetes API (typically not manually created).